Software Requirements Specification

for

QtSE

Version 1.0 approved

Prepared by Will Frank
Tracy King
Dustin Shiozaki
Samuel Teare
Corin Thummel
Brayden Wright

DeathStar IT

October 2017

Table of Contents

1. Introduction

1.1 Purpose

This document specifies the software requirements for QtSE. QtSE is a sprite editor that will be developed using Qt Creator.

1.2 Document Conventions

Font information:

- Any changes made after version 1.0 will appear in red font.

1.3 Intended Audience

The intended audience for this document is primarily developers that wish to create a sprite editor with a basic set of features. It is, however, also useful for both testers that need to know what features they should expect to test as well as users interested in the underlying workings of QtSE.

The document first gives a brief introduction to QtSE and what a reader should expect from the product. After which, usage of third party frameworks/interfaces is detailed, and software features thereafter.

1.4 Product Scope

QtSE is a free sprite editor with the goal of being easy to use. QtSE consists of a pleasing graphical user interface with all the basic supporting features of a simple sprite editor. These features allow the user to effectively create sprites and animated gifs of these sprites.

1.5 References

Existing tools:

Piskel	<https://www.piskelapp.com/>
Aseprite	<https://www.aseprite.org/>
GraphicsGale	<https://graphicsgale.com/us/>
GrafX2	<http://pulkomandy.tk/projects/GrafX2>
Spriter	<https://brashmonkey.com/>
Pickle	<http://www.pickleeditor.com/>

2. Overall Description

2.1 Product Perspective

QtSE is a self-contained project created for educational purposes, specifically as a means to become familiar with the Qt development environment. Its use is not limited to this purpose alone and is free to be use by anyone who desires to do so.

2.2 Product Functions

The basic sprite editor functions provided by QtSE includes:

- Provide a canvas of user specified dimensions.
- Allow the user to draw on the created canvas.
- Display a preview of the user's work.
- Save and load the project.
- Export the project to a GIF.
- Export the project to a sequence of PNGs or as a PNG sprite sheet.
- Ability to use various drawing tools (specifics detailed further in the document)
- Ability to undo and redo changes

Animation related functions provided by QtSE includes:

- Add and remove animation frames.
- Animate the project preview area if more than one frame exists.
- Allow the user to specify a frame rate for the animation.
- Export the project as an animated GIF.

2.3 User Classes and Characteristics

Our product only caters to one general user class. They should have at least some experience with GUIs, don't necessarily need any experience with sprite editors or drawing programs, and need no technical expertise. If a user exceeds any of these requirements they will likely be able to make better use of our program, but they are not being specifically catered to.

2.4 Operating Environment

QtSE runs on all traditional desktop operating systems as long as they have at least a desktop environment, mouse with one button, and a keyboard. There are no known conflicts with pre-existing software.

2.5 Design and Implementation Constraints

All text contained in QtSE will be in English.

2.6 User Documentation

User documentation comes in the form of simple tooltips when hovering over a UI element. Since the target users of the end product are expected to have some basic experience with a painting program, this form and amount of documentation is expected to be sufficient.

2.7 Assumptions and Dependencies

QtSE will be developed using Qt v5.9.1. QtSE will be using FreeImage for the image handling.

3. External Interface Requirements

3.1 User Interfaces

The user interface will contain a file menu which will have the following options: load sprite, save sprite, export sprite. There will be additional toolbar icons which will include the following options: bucket fill, line tool, square tool, circle tool, dithering tool, color pallet tool, rectangle selection tool. The selection tool will allow the user to cut, copy and paste pixels. After an area is selected it can be edited using hotkeys or the appropriate functions can be accessed via the edit menu. A section of the gui will be dedicated to creating frames. A slider will allow the user to select the number of frames, and a maximum of 24 frames will be available.

3.2 Hardware Interfaces

Our source code will be provided which can be compiled on Windows, Linux, and Macintosh operating systems.

3.3 Software Interfaces

The program will compile on any OS that supports Qt 5 and C++11 must be installed. We will be using Qt version 5.9.1 with MSVC2015.

4. System Features

QtSE's notable features are detailed below.

4.1 Set the Frame Size in Pixels

4.1.1 Description and Priority

The user will have the ability to set the size for the sprite. The available frame sizes are 8x8, 16x16, 32x32, 64x64, and 128x128. Pixels will be sized in powers of two. This is a high priority.

4.1.2 Stimulus/Response Sequences

To set the Frame Size: User clicks on the New option under the File menu button. A drop down menu that includes the available frame sizes is displayed. The user selects a size option. The frame size is adjusted to the selected size.

4.1.3 Functional Requirements

The default frame size is 32x32.

4.2 Adjust Number of Frames

4.2.1 Description and Priority

The user will have the option to add frames to the Sprite via the add frame button. The user will be able to remove frames via the delete frame button. This is a high priority.

4.2.2 Stimulus/Response Sequences

To add a frame: User clicks on the add frame button. A new frame will be added to the end of the list of frames in the Sprite.

To remove a frame: User clicks on the remove frame button on the frame they desire to remove. That frame is removed from the list and frames that come after that frame are shifted to close the gap.

4.2.3 Functional Requirements

If there is only 1 frame, the remove frame button will be disabled and not allow the last frame to be removed. Once there is more than one frame, all frames will have the remove frame button enabled.

4.3 Modify Sprite's Pixels

4.3.1 Description and Priority

A user will be able to modify Sprite by adding pixels, removing existing pixels or changing the color of existing pixels. These modifications will be handle via the editing tools provided. These tools include pen tool, mirror pen tool, eraser tool, bucket fill tool, color fill tool, rectangle tool, circle tool, rectangle select tool, and dithering tool. The desired color can be selected by the color selection tool. This is a high priority.

4.3.2 Stimulus/Response Sequences

To select a color: User will click on the color selection tool. The available color options will be displayed. User selects a desired color. That color becomes the current selected color.

To use the pen tool: User will click on the pen tool icon or press the hotkey. The pen tool will become the selected tool. User clicks on any pixel of the canvas. That pixel is changed to the selected color. User may click, hold, and drag with the pen tool. All the pixels that the click, hold, and drag touched are changed to the selected color.

To use the mirror pen tool: User will click on the mirror pen tool icon or press the hotkey. The mirror pen tool will become the selected tool. User clicks on any pixel of the canvas. That pixel and the pixel that has the same y and equal x distance from the vertical center of the frame are changed to the selected color. User may click, hold, and drag with the mirror pen tool. All the pixels that the click, hold, and drag touched along with the pixels that have the same y and equal x distance from the vertical center of the frame are changed to the selected color.

To use the eraser tool: User will click on the eraser tool icon or press the hotkey. The eraser tool will become the selected tool. User clicks on any pixel of the canvas. That pixel is changed to have no color and be transparent. User may click, hold, and drag with the eraser tool. All the pixels that the click, hold, and drag touched are changed to the selected color.

To use the bucket fill tool: User will click on the bucket fill tool icon or press the hotkey. The bucket fill tool will become the selected tool. User clicks on any pixel of the canvas. That pixel and all other pixels that have the same color and can be connected back to the original pixel via pixels of the same color that are either above, below, left or right of that pixel are changed to the selected color.

To use the color fill tool: User will click on the color fill tool icon or press the hotkey. The color tool will become the selected tool. User clicks on any pixel of the canvas. That pixel and all other pixels with the same color are changed to the selected color.

To use the rectangle tool: User will click on the rectangle tool icon or press the hotkey. The rectangle tool will become the selected tool. User clicks on any pixel of the canvas, holds and drags to any other pixel of the canvas. A transparent rectangle connecting the two pixels is displayed while the mouse button is being held. Upon release of the mouse button, the pixels making up the rectangle connecting the two pixels selected by the user are changed to the selected color.

To use circle tool: User will click on the circle tool icon or press the hotkey. The circle tool will become the selected tool. User clicks on any pixel of the canvas, holds and drags to any other pixel of the canvas. A transparent circle/oval centered between the two points and with a vertical diameter spanning the vertical distance between the two pixels and a horizontal diameter spanning the horizontal distance between the two pixels is displayed. Upon release of the mouse button, the pixels making up the circle/oval that are being covered by the transparent circle are changed to the selected color.

To use the rectangle select tool: User will click the on the rectangle select tool icon in the toolbar or press the hotkey. The rectangle select tool will become the selected tool. User clicks on any pixel in the canvas, holds and drags to any other pixel of the canvas. A transparent rectangle connecting the two points is displayed while the mouse button is being held down. Upon release of the mouse button, a visual indicator in the shape of a rectangle that connects the starting and end points is left. This visual indicator shows which pixels on the canvas were selected.

To use the dithering tool: User will click on the dithering tool or press the hotkey. The dithering tool will become the selected tool. User clicks on any pixel of the canvas. That pixel is changed to the selected color. Any additional clicks or click, hold, and drag will change every other pixel or pixels that are located in any diagonal direction for the original pixel to the selected color.

To select color: User clicks on the color preview. The color select dialog is displayed. The user clicks on a desired color. That color becomes the selected color.

4.3.3 Functional Requirements

The default selected tool is the pen tool. The default selected color is black.

4.4 Sprite Animation Cycle Preview

4.4.1 Description and Priority

A Sprite Animation Cycle preview box will be displayed while the user is editing the sprite. The sprite animation cycle preview box will have a spin box that allows the user to adjust the frames per second (fps) at which the sprite animation cycle preview box cycles through the Sprite's frames. The sprite animation cycle preview box will also have the option of viewing the animation at the actual size of the sprite or at a zoomed in size. This is a high priority.

4.4.2 Stimulus/Response Sequences

To adjust the frame rate: User clicks the number in the FPS Spin Box. The FPS Spin Box will accept the user's typed input. User types the desired number of fps at which they want to preview the animation cycle. The animation cycle preview box will begin cycling the animation at the new frame rate.

To adjust the size of the playback: User clicks the preview size button. The sprite animation cycle preview box switches between 1x, or actual size, and full, or fill the preview box. The preview size button text will change to the current size selection (either 1x or full).

4.4.3 Functional Requirements

The default frame rate for the animation cycle preview will be 12 fps. The FPS spin box will have a minimum value of 0 fps and a maximum value of 24 fps. Any value less than 0 entered by the user will be changed to 0. Any value greater than 24 will changed to 24. The default size of the preview is full.

At 0 fps the sprite animation cycle preview box will display the current selected frame of the sprite animation and will not cycle through the other frames of the sprite.

4.5 Save and Load Existing Projects

4.5.1 Description and Priority

In the File drop down menu the user will be able to save or load an existing project. QtSE will use a default file type of .ssp This is a high priority.

4.5.2 Stimulus/Response Sequences

To Save a Project: User will select “Save As” from the File drop down menu. A file explorer box will open allowing the user to select a correct destination and file name. The default extension will be .ssp.

To Load a Project: User will select “Load Existing” from the File drop down menu. A file explorer box will open allowing the user to select a correct file source. QtSE will support .ssp extension.

4.5.3 Functional Requirements

The system default for saving and loading files will be an .ssp extension. If a user attempts to load an incompatible file type an error box will pop up notifying the user of it.

4.6 Export Sprite as an Animated GIF

4.6.1 Description and Priority

The user will be able to select an option from the File drop down menu to export and save the sprite in a gif format file. This is a medium priority.

4.6.2 Stimulus/Response Sequences

The user will select the option from the File drop down menu, a pop up box will allow the user to select a location and filename to save the export file. The default format will be GIF.

4.6.3 Functional Requirements

LIB-1: FreeImage

4.7 Duplicate Animation Frames

4.7.1 Description and Priority

User will have the ability with to hover over a frame and create a duplicate. This will be ideal if only minor changes are required for the next frame. This is a medium priority.

4.7.2 Stimulus/Response Sequences

To duplicate a frame: User clicks on duplicate icon on the desired frame to be duplicated. The duplicate frame is added as the next in sequence.

4.8 Color Palette of Color History

4.9.1 Description and Priority

On the user interface, there will be a grid that shows previously used colors, to enable to reuse a color again.

4.9.2 Stimulus/Response Sequences

Color pallette on the interface, with color previews of previously used colors, selected with a mouse click.

4.9 Hotkeys/Keyboard shortcuts

4.10.1 Description and Priority

User will be able to use various keystrokes for functionality in the program. Hotkeys for keyboard shortcuts will be set up, allowing the user to switch between different drawing tools and quick access to other functions.

4.10.2 Stimulus/Response Sequences

To use hotkeys: The user will pressed the hotkey associated with the desired drawing tool. That drawing tool will become the selected drawing tool.

To use keyboard shortcuts: The user will press the keyboard shortcut associated with the desired function. That function will be processed.

4.10 Undo/Redo

4.11.1 Description and Priority

User will be able to undo/redo the last action. This option will appear in the toolbar.

4.11.2 Stimulus/Response Sequences

To use undo: The user clicks on the undo option in the Edit drop down menu. The current frame is reverted to its state before the last action was performed.

To use redu: The user clicks on the redo option in the Edit drop down menu. The frame is updated to its state before the last undo action was performed.

4.11.3 Functional Requirements

There will be a finite number of past actions the user will be able to undo. This will then extend to a finite number of actions the user will be able to redo. This number is TBD.

4.11 Export Sprite as PNG Files or a PNG Sprite Sheet

4.12.1 Description and Priority

Allow the user to export their sprite as either a sequence of PNG images or a traditional style sprite sheet saved as a single PNG image.

4.12.2 Stimulus/Response Sequence

To export current frame as PNG: The user clicks on Export under the File menu. A drop down menu is displayed with different export options. User clicks on Current Frame. A select save destination window is displayed. The user selects the save destination. The current frame is saved as a PNG file to the selected destination.

To export all frames as PNG: The user clicks on Export under the File menu. A drop down menu is displayed with different export options. User clicks on All Frames. A select save destination window is displayed. The user selects the save destination. All frames are saved as individual PNG files to the selected destination.

To export Sprite as Sprite Sheet (PNG): The user clicks on Export under the File menu. A drop down menu is displayed with different export options. User clicks on Sprite Sheet. A select save destination window is displayed. The user selects the save destination. The frames of the Sprite are saved in order as a single PNG file to the selected destination.

4.12.3 Functional Requirements

REQ-1: FreeImage library

5. Other Requirements

Appendix A: Glossary

QtSE	- Name of the program described in this SRS. The correct pronunciation of QtSE is "cutesy".